

Primero de todos creamos un proyecto en android studio, con la actividad vacía.

Y empezamos con la primera actividad

Bueno se puede utilizar cualquier “Layout ” para ajustar los botones, por Ejemplo yo he utilizado un “constraint layout” dentro un “LinearLayout” y dentro tiene dos botones, y luego bajo de linear layout he puesto un textView Vacío que luego cuando volvamos de una actividad diferente, recibiremos un mensaje desde la actividad anterior y le mostramos en el textView que habremos puesto vacío.

Así esta el código:

“ConstraintLayout”:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

LinearLayout, y dos botones que van a llamar a otra actividad:

```
<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="339dp"
    android:layout_height="67dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="32dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:gravity="center_horizontal"
    android:orientation="horizontal"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <Button
        android:id="@+id/buttonActividad1"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Actividad 1" />

    <Button
        android:id="@+id/buttonActividad2"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Actividad 2" />

</LinearLayout>
```

y al final un textView Vacío para la razón que he comentado anteriormente.

```

<TextView
    android:id="@+id/textView"
    android:layout_width="332dp"
    android:layout_height="82dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:text=""
    android:textSize="30dp"
    android:textColor="@android:color/black"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Creamos Otras dos Actividades y pondremos un “text view” dentro de la actividad correspondiente diciendo que es la primera o la segunda.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Actividad1">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Actividad 1"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Actividad2">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Actividad 2"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

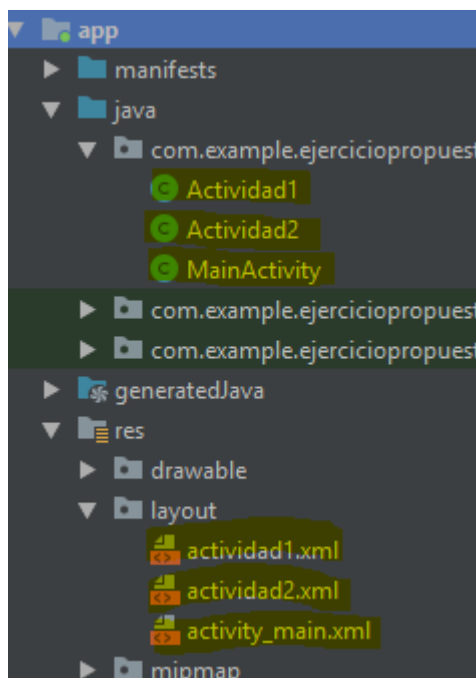
```

Ya que nosotros podemos fijar en nuestro código “xml” que tenemos puesto un `tools:context` a una clase java que no tenemos hasta ahora.

```
tools:context=".Actividad2">
```

```
tools:context=".Actividad1">
```

Así que creamos estas dos clases también.
Al final la estructura se va a quedar así:



Muy Importante:

Cuando creamos una nueva clase, hay que poner en “AndroidManifest.xml” por que si no, No va a funcionar la Aplicación.

Vamos a agregarle:



Ya tenemos, estructurado nuestra programa, ahora hay que programar para que funcione lo que se debe de hacer la aplicación.

Vamos a nuestra clase principal “MainActivity.java”

Obtenemos el textView, el que habíamos dejado vacío.

Y creamos un objeto “Bundle” que nos trae el texto que trae “Intent” desde una actividad a otra.

Y si es distinto que nulo, pues sacaremos el texto en el textView Vacío que hemos creado anteriormente.

Creamos dos botones, Y con la ayuda de la clase “Intent” podremos acceder a nuestra otra actividad desde principal o volver de una actividad a principal.

Así que esta implementado que si damos al primera actividad vuelve con la primera actividad y asigna el valor (que trae desde la primera actividad) al textView de esta actividad.

Y igualmente con el segundo botón.

Importante: Cuando creamos “Intent”, Siempre hay que iniciar esta actividad porque si no, no iría.

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TextView v = findViewById(R.id.textView);
        Bundle b= getIntent().getExtras();
        if(b != null)
            v.setText(b.getString( key: "Texto"));

        Button b1 = findViewById(R.id.buttonActividad1);
        Button b2 = findViewById(R.id.buttonActividad2);

        b1.setOnClickListener((v) -> {
            Intent i = new Intent( packageContext: MainActivity.this, Actividad1.class);
            startActivity(i);
        });

        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent( packageContext: MainActivity.this, Actividad2.class);
                startActivity(i);
            }
        });
    }
}
```

Y para que se termine la aplicación cuando este en la actividad principal: Necesitamos poner 4 código de líneas para terminar la aplicación, como pulsamos el botón para atrás en la actividad principal, debe de cerrar la aplicación. (Se puede programar cerrando preguntando “Salir o cancelar” pero no lo necesitamos).

```
@Override
public void onBackPressed() {
    Intent i = new Intent(Intent.ACTION_MAIN);
    i.addCategory(Intent.CATEGORY_HOME);
    i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(i);
}
}
```

Hago un “Toast” (Básicamente un mensaje, que estoy en la primera actividad y punto) mostrandolo con el método show(),

Y aquí también hay que implementar en “OnBackPressed()” para que si le damos a atrás nos lleve a la actividad principal con un intent, y fijamos que le hemos puesto un valor extra, este valor extra va a pillar el “Bundle” de “MainActivity” y ya no será nulo, y entrará en el “if” y mostrará el mensaje en el textView de la actividad principal. Y obviamente si no iniciamos la actividad no podremos conseguir lo que queremos.

```
package com.example.ejerciciopropuesto3activities;

import ...

public class Actividad1 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.actividad1);

        Toast t = Toast.makeText(getApplicationContext(), text: "Actividad Secundario 1 iniciado desde Actividad principal", Toast.LENGTH_LONG);
        t.show();
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent i = new Intent( packageContext: Actividad1.this, MainActivity.class);
        i.putExtra( name: "Texto", value: "Volviendo de Actividad 1 -> Activity1 Finalizado");
        startActivity(i);
    }
}
```

Y igualmente lo que hemos hecho con nuestra primera actividad, así mismo haremos con la segunda pero cambiando el mensaje “Desde la segunda” obviamente que tenga sentido.

```
package com.example.ejerciciopropuesto3activities;

import ...

public class Actividad2 extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.actividad2);

        Toast t = Toast.makeText(getApplicationContext(), text: "Actividad Secundario 2 iniciado desde Actividad principal", Toast.LENGTH_LONG);
        t.show();
    }

    @Override
    public void onBackPressed() {
        super.onBackPressed();
        Intent i = new Intent( packageContext: Actividad2.this, MainActivity.class);
        i.putExtra( name: "Texto", value: "Volviendo de Actividad 2 -> Activity2 Finalizado");
        startActivity(i);
    }
}
```

Probamos nuestro programa y disfrutamos de lo que aprendemso.

Gracias!